Human Activity Recognition Based on Machine Learning and Smartphone Sensors

ZhiHan Zhao*

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China, 650500

*Corresponding author: Z3443498185@163.com

Abstract. In recent years, the importance of Human Activity Recognition (HAR) technology has continued to grow. This study utilizes smartphone-embedded Inertial Measurement Units (IMUs) to collect real-world accelerometer and gyroscope data, constructing a multi-dimensional activity analysis framework. Through systematic evaluation of time-domain, frequency-domain, and time-frequency domain feature extraction methods, combined with SMOTE data balancing and PCA dimensionality reduction techniques, a hierarchical classification architecture was designed. This framework integrates machine learning models including logistic regression, Support Vector Machine (SVM), Random Forest, XGBoost, and KNN. The results demonstrate differentiated characteristics of classifiers in activity recognition tasks. By dynamically selecting classifier combinations adapted to different activity features, the system achieves flexible optimization of classification strategies. Validation reveals that the integration of time-frequency domain features with dimensionality reduction effectively enhances computational efficiency. The proposed hierarchical framework provides technical support for lightweight HAR systems on smartphone platforms, with its multi-feature domain fusion strategy being extendable to daily activity monitoring and health assessment scenarios. This research offers methodological references for human behavior understanding in wearable devices, emphasizing the critical role of feature engineering and classifier co-optimization in practical applications.

Keywords: Human Activity Recognition, Smartphone Sensors, Machine Learning.

1. Introduction

Human Activity Recognition (HAR) is defined as the technology that classifies temporal human actions through discrete sensor measurement data [1]. This technology collects physical quantities such as acceleration, angular velocity, and geographic coordinates to construct a mapping model from multidimensional signals to behavior categories, showing broad application prospects in the field of the Internet of Things (IoT) [2]. Currently, deep learning algorithms and shallow learning algorithms are the two mainstream methods for studying HAR problems [3]. Deep learning-based HAR models can automatically extract significant features for recognition through various filters [4]. However, due to their black-box nature, reliance on large datasets, and high computational costs, shallow machine learning methods remain the preferred choice when the training dataset is small or rapid training is required [5].

In the research and application of HAR technology, based on the different methods of data acquisition, it is mainly divided into four implementation approaches: visual signals, WiFi signals, audio signals, and sensor signals [6]. For the precise capture of fine-grained human activities, these four methods have each demonstrated high potential but also have their respective limitations. Visual signal-based HAR is susceptible to environmental influences such as lighting and angles, requires high deployment and cost for camera setups, and is difficult to deploy; WiFi signal-based HAR mainly analyzes public WiFi hotspot data for recognition, which is easy to deploy but difficult to detect and requires specific distances between the detector and the WiFi hotspot; audio signal-based HAR can use the microphone and speaker on a smartphone to form a small sonar system, analyzing sound signals to recognize human activities, but it is easily affected by ambient noise. In contrast, sensor signals are more closely related to human activities, are less affected by environmental factors, and do not require specific deployment environments. Thanks to the development and popularization

of chip technology and IoT, an increasing number of researchers are using wearable devices for studies. Ermes et al. utilized a hybrid classifier combining tree structures and artificial neural networks (ANN) to recognize nine daily and sports activities. Through 12-fold leave-one-subject-out cross-validation, the classification accuracy for nine daily and sports activities using supervised and unsupervised data was 89% [7]. Margarito et al. compared the recognition performance of template matching and statistical learning classifiers to identify eight common sports activities using a triaxial accelerometer worn on the user's wrist [8]. Wang et al. used accelerometers, gyroscopes, and other sensors, employing machine learning methods such as random forests and support vector machines for classification, achieving an accuracy of up to 95% in user authentication by recognizing keystroke characteristics [9].

The focus of this study is on using inertial sensors embedded in smartphones (such as accelerometers and gyroscopes) combined with machine learning (ML) tools to accurately recognize human activities [10]. This paper introduces a hierarchical human activity recognition framework based on smartphone sensors, aiming to advance the development of HAR technology based on smartphone sensors and provide guidance for its optimization in practical applications. The primary objective of this study is to systematically evaluate the performance of different machine learning classifiers, explore the impact of feature extraction methods on classification effectiveness, and analyze the misclassification behaviors of models in practical applications. This aims to provide theoretical support and methodological guidance for constructing efficient, robust, and practical Human Activity Recognition (HAR) systems. The three main contributions of this paper are as follows:

(1) We systematically evaluated the performance of various machine learning classifiers in HAR tasks, including both shallow learning and deep learning algorithms. By comparing the performance of different classifiers on different datasets, we revealed the strengths and limitations of each classifier and proposed optimization strategies. This contribution provides a theoretical basis for researchers to choose appropriate classifiers in practical applications, especially in scenarios with small training datasets or the need for rapid training.

(2) We comprehensively explored the impact of time-domain, frequency-domain, and time-frequency-domain combined features on HAR tasks for the first time, and quantitatively analyzed the contribution of different feature subsets to the classification of various actions through ablation experiments. This contribution not only reveals the importance of feature selection for classification performance but also provides optimization directions for building more efficient HAR systems, particularly in recognizing short-term, low-frequency posture transition actions.

(3) We proposed a model performance analysis method based on confusion matrices, by calculating the misclassification rates of various actions to identify behavior categories with poor classifier performance and deeply analyze the underlying reasons. This contribution not only provides specific guidance for model optimization but also offers methodological support for enhancing the practicality and robustness of HAR systems, especially in handling complex and diverse real-world activity scenarios.

2. Methodology

The workflow of our activity recognition model is illustrated in Figure 1. Prior to model fitting, the label types in the dataset were expanded from 6 to 12, and data preprocessing was simultaneously conducted, including Principal Component Analysis (PCA), oversampling techniques, and data normalization. The processed data was labeled as stratified data and selected data, corresponding to linear classifiers and non-linear classifiers, respectively. The former employs a logistic regression model, while the latter utilizes various classic machine learning models, such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN). Finally, we performed a comprehensive performance analysis and visualized the results using a confusion matrix.

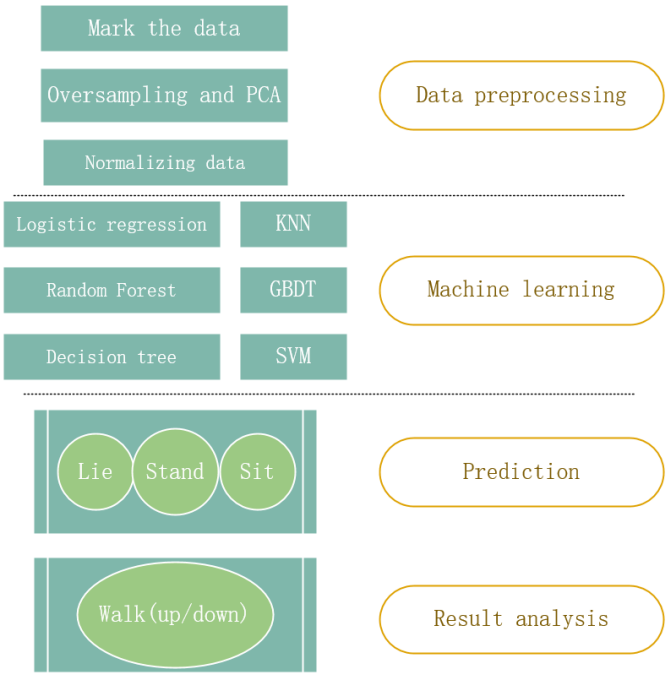


Figure. 1. The structure of model

2.1. Description of dataset

The dataset used in this experiment is derived from the comprehensive UCI HAR Dataset, which can be accessed via https://www.kaggle.com/datasets/ermitw/ucihar. This dataset was experimentally measured and processed by researchers at UCI. The subjects of the experiment were 30 volunteers aged between 19 and 48 years. Each individual performed six activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying down. Additionally, each person had a smartphone securely fastened to their waist. Using the smartphone's built-in accelerometer and gyroscope, 3-axis linear acceleration and 3-axis angular velocity were captured at a constant frequency of 50 Hz. The obtained dataset has been randomly divided into two groups, with 70% of the volunteers selected to generate the training data and 30% selected to generate the test data.

2.2. Data preprocessing

2.2.1 Time Domain and Frequency Domain

We utilized the UCI-HAR dataset, which was constructed from recordings of daily activities (ADLs) performed by 30 volunteers carrying waist-mounted smartphones embedded with inertial sensors. This dataset provides 561 features, including 265 time-domain features, 289 frequency-domain features, and angular data. The activities in the dataset include walking, walking upstairs, walking downstairs, sitting, standing, and lying down. These data belong to time-series data, and there are transitional behavioral changes between these labels. We also considered these transitions as a special type of label. Consequently, we relabeled the data as detailed in Table 1.

Table 1. The remarked labels

labels	the changes of activities
1	Walking
2	Walking upstairs
3	Walking downstairs
4	Sitting
5	Standing
6	Laying
7	From Standing to Sitting

8	From Sitting to Standing
9	From Sitting to Laying
10	From Laying to Sitting
11	From Standing to Laying
12	From Laying to Standing

2.2.2 SMOTE

SMOTE (Synthetic Minority Oversampling Technique) is an oversampling method designed for imbalanced datasets. It generates synthetic samples to balance the minority class, thereby enhancing the classification model's ability to recognize the minority class. In this paper, due to the short duration of transitional activities, some of these activities do not even span a full window. Consequently, the number of samples that meet the window size requirement is actually very small, insufficient to provide meaningful information for generating a better training model. Based on this, we employ the SMOTE algorithm to oversample the minority class samples. This is achieved by replicating samples from the minority class in the training dataset before fitting the model. This approach ensures a balanced class distribution without introducing any additional information to the model. The principle is as follows:

1. Selecting a Sample: Randomly select a sample x_i from the minority class.

2. Finding Neighbors: Calculate the $ k $ nearest neighbors of x_i .

3. Generating Synthetic Samples: Randomly select a neighbor x_j and generate a new sample by interpolating between x_i and x_j :

$$x_{\text{new}} = x_i + \lambda \cdot (x_j - x_i) \tag{1}$$

where λ is a random number in the range [0,1].

4. Repeating: Repeat the process until the number of minority class samples is balanced with the majority class.

2.2.3 PCA

PCA (Principal Component Analysis) is an unsupervised dimensionality reduction technique that projects high-dimensional data into a lower-dimensional space through linear transformation, preserving the maximum variance information in the data. PCA maps the original high-dimensional data to a lower-dimensional space via linear transformation while retaining the main features and information of the data. The new variables obtained through this transformation are called principal components, which are linear combinations of the original variables and are uncorrelated with each other. Our data contains 561 features, which is a large number. We reduce the dimensionality to 128 features to extract key features and patterns, thereby improving the model's performance and accuracy. The principle is as follows:

1. Standardization: Remove the mean and standardize the variance of the original data.

2. Calculating the Covariance Matrix: Reflect the correlations between features.

3. Eigenvalue Decomposition: Solve for the eigenvalues and eigenvectors of the covariance matrix.

4. Selecting Principal Components: Arrange the eigenvectors in descending order of eigenvalues and select the top k eigenvectors as principal components.

5. Projecting the Data: Map the original data to the new coordinate system:

$$Z = X \cdot W \tag{2}$$

where W is the principal component matrix.

2.3. Hierarchical Model

Considering the switching between human activities and the differences in activities themselves, a classification is constructed using a logistic regression model with labels 1-6 and labels 7-12 for binary classification. Considering the significant differences between walking behavior and other activities, a logistic regression model is used to implement binary classification of labels 1-3 and 4-6. The specific structure of the Hierarchical Model is shown in Figure 2.

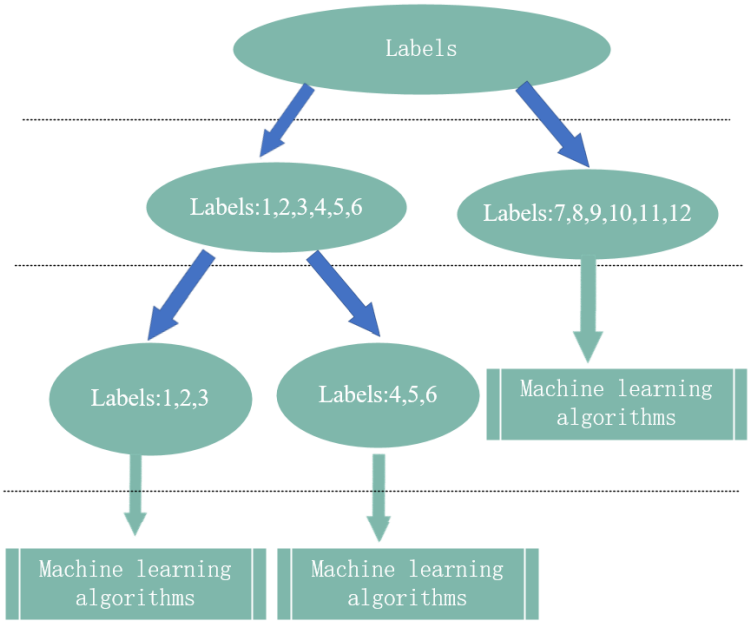


Figure.2 Hierarchical implementation of binary classification

2.4. Machine Learning Models

In this study, we systematically evaluated and compared various machine learning algorithms to identify the optimal model for human activity recognition.

2.4.1 Support Vector Machine (SVM)

The fundamental principle of the Support Vector Machine lies in its capability of "dimensionality transformation." This technique enables the conversion of a low-dimensional, linearly inseparable space into a high-dimensional, linearly separable space. Through this transformation, regardless of whether the initial sample points are linearly separable, approximately linearly separable, or nonlinearly separable, they can be accurately classified using a precisely defined "hyperplane." This characteristic makes SVM particularly effective in handling complex classification problems.

2.4.2 K-Nearest Neighbors Algorithm (KNN)

As a representative "lazy" learning algorithm, the K-nearest neighbor algorithm performs both model construction and prediction simultaneously. It operates by comparing sample similarities between the training dataset and the testing dataset, subsequently identifying the k most similar samples to make predictions for unknown data points. The selection of the k value is crucial: an excessively small k value may result in model overfitting, while an overly large k value could lead to underfitting. To address this challenge, we implemented multiple cross-validation techniques in our experiments to determine the optimal k value that best fits our data characteristics.

2.4.3 Classification and Regression Tree (CART)

The decision tree model is fundamentally based on "if...then..." conditional relationships, offering an intuitive and clear structure. Through the calculation of information entropy and information gain rate corresponding to human activity features, we can effectively analyze the contribution probability

of each feature and identify the optimal path leading to the correct classification label. This approach provides both interpretability and effectiveness in feature analysis.

2.4.4 Extreme Gradient Boosting (XGBoost)

The XGBoost algorithm employs an iterative approach to train decision trees, where the residuals from the previous tree serve as the training target for the subsequent tree, thereby progressively reducing prediction errors. Its core concept involves constructing new decision trees along the negative gradient direction of the loss function, continuously refining the model's performance. As an ensemble learning algorithm, XGBoost effectively combines multiple weak classifiers with corresponding weights, ensuring both high accuracy and robustness in predictions.

2.4.5 Random Forest (RF)

Compared to single decision tree models, Random Forest utilizes an ensemble of multiple decision trees, with each tree trained on randomly selected samples. The feature selection at each node is also randomized, making the model less susceptible to overfitting. Additionally, Random Forest offers a balanced computational cost and maintains interpretability, making it particularly valuable for feature selection tasks. These advantages contribute to its widespread application in various machine learning scenarios.

3. Results and discussion

3.1. Model performance analysis

The large data prediction model for the user's electricity consumption is implemented in the Clementine software. In the same classification task, different classifiers are used for model fitting, and the performance of different classifiers is evaluated and compared. At the same time, considering the uneven distribution of samples in the test data set, we use weighted average when calculating performance indicators, which can better reflect the contribution of different categories to the overall performance. The model indicators use precision, recall, F1-score, and accuracy, and their calculation formulas are as follows:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

TP means true positive, TN means true negative, FN means false negative, and FP means false positive.

For classification tasks 3, 4, and 5, we use model accuracy as a measure and compare different classifiers to find that support vector machines have the best fitting effect. We choose support vector machine as the classifier for fitting and provide relevant parameters. The detailed results are presented in Table 2.

Table 2. Classification Results

Classification task	Precision	Recall	F1-score	Accuracy
Task 1. (logistic regression)	0.93	0.79	0.85	0.7889
Task 2. (logistic regression)	1.00	1.00	1.00	1.0000
Task 3. (Regression Tree)	0.73	0.73	0.73	0.7315
Task 3. (Random Forest)	0.89	0.89	0.89	0.8928

Task 3. (XGBoost)	0.88	0.88	0.88	0.8778
Task 3. (SVM)	0.94	0.94	0.94	0.9416
Task 3. (KNN)	0.86	0.86	0.86	0.8602
Task 4. (Regression Tree)	0.81	0.80	0.81	0.8045
Task 4. (Random Forest)	0.91	0.91	0.91	0.9105
Task 4. (XGBoost)	0.92	0.92	0.92	0.9167
Task 4. (SVM)	0.95	0.95	0.95	0.9470
Task 4. (KNN)	0.90	0.89	0.89	0.8933
Task 5. (Regression Tree)	0.34	0.33	0.34	0.3310
Task 5. (Random Forest)	0.41	0.42	0.42	0.4225
Task 5. (XGBoost)	0.40	0.40	0.40	0.4014
Task 5. (SVM)	0.45	0.44	0.44	0.4366
Task 5. (KNN)	0.36	0.36	0.36	0.3592

3.2. Ablation Experiment Analysis

In this section, ablation experiments in time domain and frequency domain are performed to evaluate the importance of each feature. If a feature is important, when we delete it, it will have a large deviation on the accuracy of the model. Modify the dataset to get two variants. It includes (1) removing the frequency-domain features in the data set, that is, deleting the frequency-domain influence (2) removing the time-domain features in the data set, that is, deleting the time-domain influence. The experimental results are shown in the table3 and table4:

Table 3. Only time domain

Classification task	Precision	Recall	F1-score	Accuracy
Task 1. (logistic regression)	0.93	0.76	0.83	0.7604
Task 2. (logistic regression)	1.00	1.00	1.00	1.0000
Task 3. (Regression Tree)	0.74	0.74	0.74	0.7374
Task 3. (Random Forest)	0.90	0.90	0.90	0.9016
Task 3. (XGBoost)	0.93	0.93	0.93	0.9312
Task 3. (SVM)	0.97	0.96	0.96	0.9644
Task 3. (KNN)	0.86	0.85	0.85	0.8513
Task 4. (Regression Tree)	0.79	0.79	0.79	0.7852
Task 4. (Random Forest)	0.91	0.91	0.91	0.9119
Task 4. (XGBoost)	0.93	0.92	0.92	0.9236
Task 4. (SVM)	0.95	0.95	0.95	0.9511
Task 4. (KNN)	0.88	0.88	0.88	0.8809
Task 5. (Regression Tree)	0.37	0.37	0.37	0.3732
Task 5. (Random Forest)	0.47	0.46	0.46	0.4577
Task 5. (XGBoost)	0.50	0.50	0.49	0.5000
Task 5. (SVM)	0.42	0.43	0.42	0.4296
Task 5. (KNN)	0.43	0.44	0.43	0.4366

Table 4. Only frequency domain

Classification task	Precision	Recall	F1-score	Accuracy
Task 1. (logistic regression)	0.93	0.78	0.84	0.7838
Task 2. (logistic regression)	1.00	1.00	1.00	1.0000
Task 3. (Regression Tree)	0.71	0.71	0.71	0.7093
Task 3. (Random Forest)	0.88	0.88	0.87	0.8750
Task 3. (XGBoost)	0.91	0.91	0.91	0.9120
Task 3. (SVM)	0.93	0.92	0.92	0.9231
Task 3. (KNN)	0.81	0.80	0.80	0.8062

Task 4. (Regression Tree)	0.58	0.58	0.58	0.5781
Task 4. (Random Forest)	0.77	0.77	0.77	0.7701
Task 4. (XGBoost)	0.80	0.80	0.80	0.7990
Task 4. (SVM)	0.81	0.81	0.81	0.8114
Task 4. (KNN)	0.70	0.68	0.68	0.6820
Task 5. (Regression Tree)	0.33	0.32	0.32	0.3239
Task 5. (Random Forest)	0.34	0.37	0.35	0.3732
Task 5. (XGBoost)	0.33	0.35	0.34	0.3450
Task 5. (SVM)	0.40	0.39	0.39	0.3873
Task 5. (KNN)	0.32	0.32	0.32	0.3169

In the feature space, we removed the frequency index while retaining the time index. Using classification task 4 as an example, the average classification accuracy with only time indicators was 89.054%. However, when constructing a model with both time and frequency characteristics, the average accuracy rose to 89.44%. These data reveal that the time feature is a crucial element in model construction. The exclusion of frequency domain indicators does negatively affect the model's overall accuracy, but incorporating both time and frequency domain indicators still yields an accuracy that is roughly 0.5 percentage points higher than using time-domain indicators alone.

By removing the time indicators and retaining the frequency indicators in the feature space, taking classification task 4 as an example again, its average classification accuracy drops to 72.812%, representing a decrease of approximately 17 percentage points compared to 89%. Considering that the accuracy of its decision tree model is only 57.81%, it is likely that this is due to the influence of extreme values. After excluding this model, the average accuracy stands at 76.56%, marking a decrease of about 12 percentage points. This demonstrates that time-domain indicators play an indispensable role in the feature space.

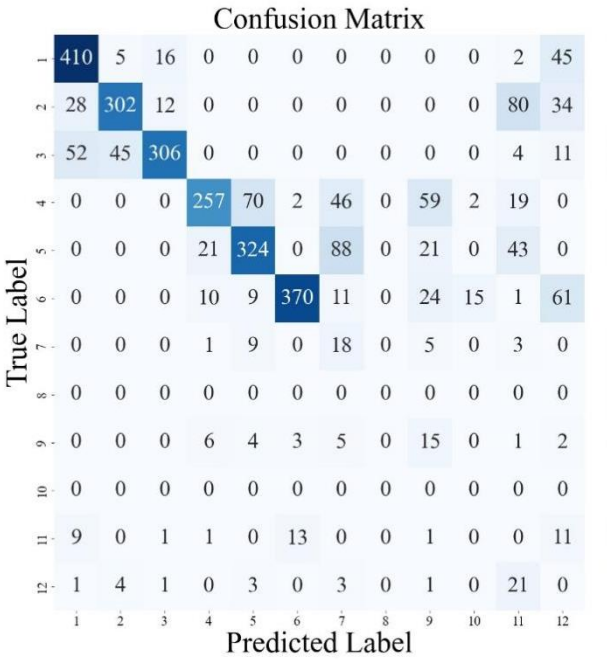
However, when time indicators are removed in classification task 3, the results do not show much difference, with some even achieving higher accuracy than when both time-frequency and frequency domain features are used. This suggests that time-frequency indicators and frequency domain indicators have different scopes of application within the feature space. For time-domain data, it can effectively capture human activities with obvious instantaneous changes or dynamic trends, such as sitting down, standing up, and lying down in task 4. After sitting down, a person remains in that position for a period of time, representing a single action that is not repeated and is discrete in nature. In contrast, the walking action in task 3 involves a gait that needs to be constantly repeated, possessing periodicity. Periodic data detected by sensors, after undergoing Fourier transform, can reveal frequency changes and distribution. Considering this property, frequency domain data has certain advantages for the classification of task 3.

In summary, due to the unique strengths of time-domain and frequency-domain features in capturing different types of information, and given the need to classify both periodic and non-periodic actions in the entire experiment, a decision was made to construct the model using time-frequency domain indicators in the feature space. However, the results of classification task 5 were not satisfactory, regardless of whether time-domain or frequency-domain indicators were used, indicating that these factors were not the cause. The possible reason is that the data for this task accounted for too small a proportion of the overall dataset, making it difficult to construct a good classification model for it.

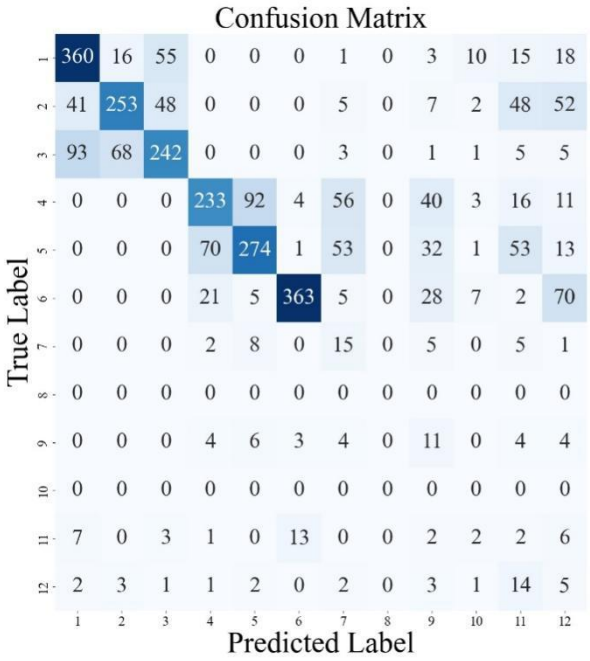
3.3. Confusion Matrix

The confusion matrices for five different machine learning algorithms are illustrated in Figure 3, providing a comprehensive view of the classification performance across various human activities. Each matrix represents the algorithm's ability to predict the true class labels (True 1-12) against the predicted labels (Predicted 1-12). Among them, Random Forest and XGBoost exhibit the highest diagonal aggregation, indicating optimal classification performance. Notable observations include the

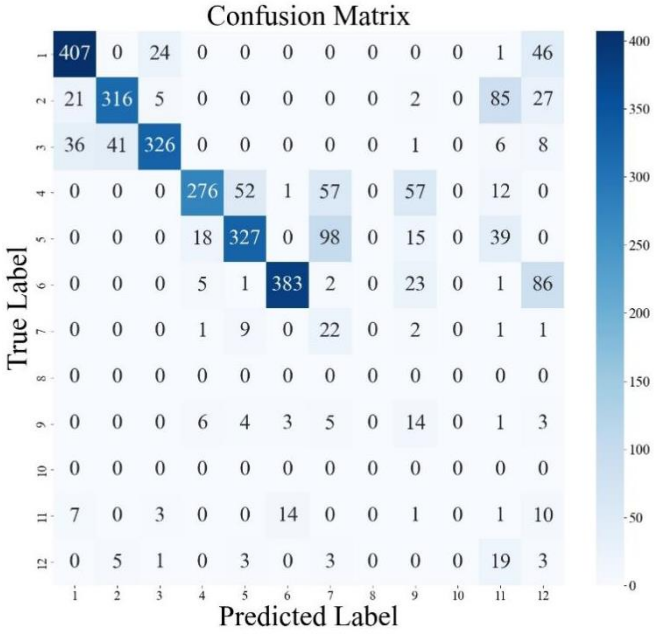
fluctuating performance of KNN in cross-class misjudgments, the formation of distinct dark blocks by XGBoost in categories, and SVM's more precise identification of individual classes.



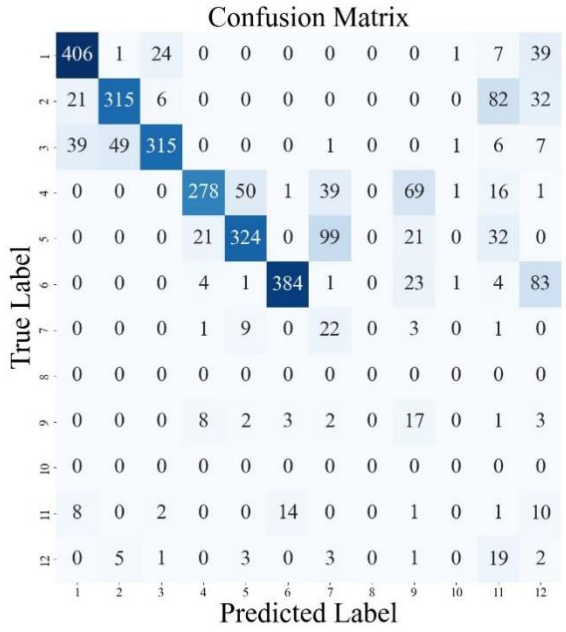
(a) KNN



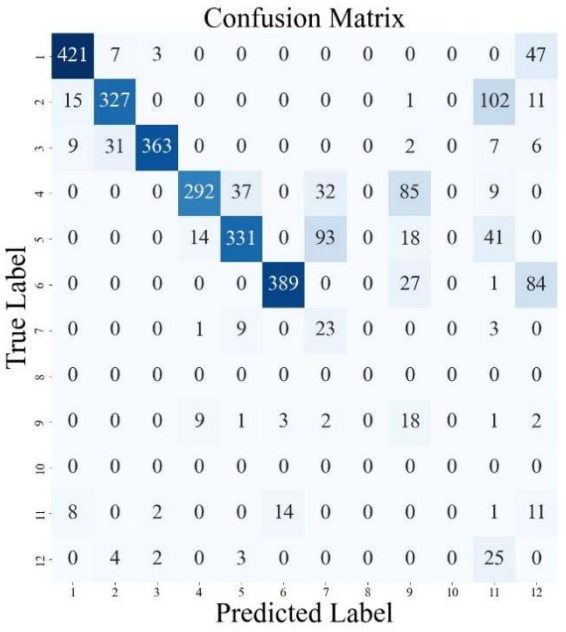
(b) Regression Tree



(c) Random Forest



(d) XGBoost



(e) SVM

Figure.3 Confusion matrices for different algorithms.

4. Conclusions

This study aims to investigate human activity recognition using machine learning models on an official dataset. To reduce the impact of unnecessary features, the research first employs PCA to simplify the data features and utilizes the SMOTE algorithm to address the issue of data imbalance. The study designs multiple machine learning classifiers and, for the first time, introduces activity labels related to transitional actions. Additionally, it categorizes activities into static and dynamic activities through a hierarchical approach, with dynamic activities further divided into continuous and transitional activities. The results indicate that the Support Vector Machine (SVM) performs the best in overall classification, followed by the Random Forest (RF). To gain a deeper understanding of the models' performance in different human activity recognition tasks, the study employs confusion matrices for evaluation. Furthermore, ablation experiments are conducted to explore the impact of time and frequency features on classification. The findings suggest that the combination of time-domain and frequency-domain features provides a comprehensive tool for addressing various challenges in human activity recognition. Future research will include collecting and constructing new datasets for experimentation and attempting to improve machine learning algorithms for better results. These studies will offer new directions for human activity recognition.

References

[1] Straczkiewicz M, James P, Onnela J P. A systematic review of smartphone-based human activity recognition methods for health research[J]. NPJ Digital Medicine, 2021, 4(1): 148.

[2] Gao G, Li Z, Huan Z, et al. Human behavior recognition model based on feature and classifier selection[J]. Sensors, 2021, 21(23): 7791.

[3] Demrozi F, Pravadelli G, Bihorac A, et al. Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey[J]. IEEE access, 2020, 8: 210816-210836.

[4] Wang X, Wang X, Lv T, et al. HARNAS: Human activity recognition based on automatic neural architecture search using evolutionary algorithms[J]. Sensors, 2021, 21(20): 6927.

[5] Gao G, Li Z, Huan Z, et al. Human behavior recognition model based on feature and classifier selection[J]. Sensors, 2021, 21(23): 7791.

[6] Sun Z, Ke Q, Rahmani H, et al. Human action recognition from various data modalities: A review[J]. IEEE transactions on pattern analysis and machine intelligence, 2022, 45(3): 3200-3225.

[7] Ermes M, Pärkkä J, Mäntyjärvi J, et al. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions[J]. IEEE transactions on information technology in biomedicine, 2008, 12(1): 20-26.

[8] Wang Y, Wu C, Zheng K, et al. Improving reliability: User authentication on smartphones using keystroke biometrics[J]. IEEE Access, 2019, 7: 26218-26228.

[9] Wang Y, Wu C, Zheng K, et al. Improving reliability: User authentication on smartphones using keystroke biometrics[J]. IEEE Access, 2019, 7: 26218-26228.

[10] Gupta S. Deep learning based human activity recognition (HAR) using wearable sensor data[J]. International Journal of Information Management Data Insights, 2021, 1(2): 100046.